# *PC-600 / PCS-620 OS Embedder™ With QNX® Momentics® Eval Development Kit*

## Copyright

Copyright 2003-Octagon Systems Corporation. All rights reserved. The contents of this document and the specifications herein may change without notice.

## Trademarks

Octagon Systems Corporation®, the Octagon logo, and OS Embedder™ are trademarks of Octagon Systems Corporation.

SanDisk® is a registered trademark of Sandisk Corporation.

QNX®, Momentics®, and Photon micro GUI®, are registered trademarks of QNX Software Systems LTD.

## Notice to user

The information contained in this document is believed to be correct. However, Octagon assumes no responsibility for any of the programs described herein, conveys no license under any patent or other right and makes no representations that the programs are free from license infringement. Octagon makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification.

***PC-600 / PCS-620 OS Embedder™ with QNX® Momentics® Eval Development Kit***
***User's Manual***
*6315(1003)*

# *Table of Contents*


PC-600 / PCS-620 OS Embedder™ With QNX® Momentics® Eval Development Kit 1
- Copyright 1
- Trademarks 1
- Notice to user 1

Table of Contents 2
- Introduction 3
- Equipment supplied 3

Getting Started 4
- PS/2 Mouse Required 4
- Licensing 4
- Loading the operating system 4
- Logging in/Password 5
- Network configuration 5
- Transferring files 6
- Parallel Port 6
- IDE / EIDE 6
- Ethernet 6
- Serial (COM) ports 6
- Real Time Clock 6
- Console driver 7
- USB 7
- PC-104 and PC-104 Plus Bus and Adapters 7
- Xpress AUDIO 7
- $I^2C$ Bus Adapter 7
- Watchdog Timer 7
- Serial EEPROM 8
- Temperature Sensor 8
- Digital I/O 8
  - Using the /dev/ezio*n* device file 9

Appendix A - Programming examples 11
- Digital I/O Examples 11
- Wdtest (watchdog timer) 11
- Testtemp (temperature sensor) 11

# Overview

## Introduction

The PC-600 / PCS-620 OS Embedder with QNX Momentics Eval, Development Kit enables you to use the PC-600 and PCS-620 Single Board Computers (SBCs) with QNX Momentics. This User's Manual describes the changes required to use the QNX Momentics operating system, and includes some example programs. For a complete description of the hardware features of the PC-600 / PCS-620 SBCs, refer to the Single Board Computer hardware User's Manual, which is also on the enclosed Octagon Products, Manuals, and Catalog CD.

## Equipment supplied

The following equipment is supplied with the Octagon Systems PC-600 / PCS-620 Linux Development Kit:

- PC-600 Single Board Computer or PCS-620 Single Board Computer
- 128 MB SODIMM
- Development Cable Kit
- Network crossover cable
- PS/2 Keyboard/Mouse "Y" cable (PC-600 Kit Only)
- PS/2 Mouse
- AT Battery
- QNX Momentics Eval Kit
- Octagon Systems QNX Demo Disk
- Octagon Products, Manuals, and Catalog CD
- Hard copy of this manual
- Preloaded 64 MB Compact Flash

# *Getting Started*

This User's Guide is intended for use with the QNX Momentics 6.2, and the Octagon System's OS Embedder Development Kit and also as a reference for configuring PC-600 or PCS-620 cards with the QNX Momentics operating system. The SBC included in the Development Kit is preloaded with the QNX Momentics operating system. If you need to reload the operating system to the factory default condition, use the instructions in the following section, "Loading the operating system." Otherwise, continue to the next section, "Logging in/Password."

Refer to the section "Loading your own QNX Image" for instructions on installing a QNX image of your own creation. Refer to QNX online or printed documentation for instruction on creating your own QNX image.

## PS/2 Mouse Required

The example QNX installation, included with the OS Embedder kit, includes the QNX Photon micro GUI. In this installation, Photon is configured to use a PS/2 Mouse as a pointing device. You must connect the PS/2 mouse when you boot the device, or you will be unable to use the system. This applies to this installation only, as Photon can be configured to use a variety of pointing devices, as desired, depending on the application.

## Licensing

This Octagon Systems Development Kit includes an evaluation copy of QNX Momentics. The QNX product is supplied for evaluation and non-commercial uses only. For all inquiries regarding licensing, contact QNX directly at www.qnx.com or 1-800-676-0566.

## Loading the operating system

The SBC card included in the Development Kit is factory loaded with the QNX Momentics operating system. If you need to reload the operating system onto the SBC's compact flash card, use the following procedure.

*Note:* Flash devices only support a limited number of writes and formats before they wear out. Plan your configurations accordingly. Use of “Industrial Grade” compact flash, such as those available from SanDisk, is recommended to improve system reliability. SanDisk Industrial Grade Compact Flash devices have been tested with Octagon Systems PC-600 and PCS-620 SBCs.

1. Insert the desired Compact Flash device into the Compact Flash socket.

2. Connect a CD ROM drive to the EIDE connector. The CD ROM may be the Master or Slave device, note, however, that there MUST be one and

only one Master device on the EIDE connector, regardless of weather or not the Compact Flash device is used in the system.

3. Insert the OS Embedder CD into the drive.
4. Establish communication with the SBC using a video display and keyboard. Refer to the PC-600 or PCS-620 CPU control card User's Manual for information on connecting a serial console.
5. Enter Setup by pressing the F2 key during BIOS POST sequence (this occurs between the memory test and boot up).
6. Configure the SBC to boot to the compact flash device (NOT THE CD-ROM)
7. Reboot the system.
8. Press the Escape ("Esc") Key, during the POST screen.
9. The SBC will prompt you for the desired boot device (as a result of pressing escape in the prevous step). Use the arrow keys to highlight the CDROM option, and select with the enter / return key. Should this prompt not appear, go back to step 7 and try again.
10. Follow the on-screen dialog to load the operating system.
11. Reboot the SBC.

## Logging in/Password

At the conclusion of the boot sequence, the screen will display the Photon login screen. In the “User Name” field enter “root”. Leave the password field blank (until you set a password, described blow).

You can log in to the SBC running QNX system in 2 ways:

1. By the keyboard and video console.
2. Via the network. You can telnet to the device’s IP address and log in.

The command `/usr/bin/passwd` allows you to change the password, or to add users (use the command `/usr/bin/passwd` for more information).

## Network configuration

The ifconfig command (tyre "use ifconfig" or refer to QNX Utilities Reference for more information) to set your netowrk configurations. There are two network ports, QNX refers to them as "en0" and "en1".

### Transferring files

Transferring files to or from the SBC is easiest when done via the Ethernet. There is an FTP client (ftp) available on the SBC to ftp from the console to an external ftp server.

# PC-600 / PCS-620 Specific Hardware Drivers

### Parallel Port

QNX Standard driver, *devc-par*, provides normal PC parallel prot functionality on the PC-600 / PCS-620. See QNX provided documentation for more information.

### IDE / EIDE

QNX Standard driver, devb-eide, provides EIDE port functionality on the PC-600 / PCS-620. See QNX provided documentation for more information.

### Ethernet

QNX Standard driver, io-net and devn-eepro.so, provides networking functionality on the PC-600 / PCS-620. See QNX provided documentation for more information.

### Serial (COM) ports

QNX Standard driver, devc-ser8250, provides serial port (RS-232/422/485) functionality on the PC-600 / PCS-620. See QNX provided documentation for more information.

For access to COM 3 and COM 4, special the following arguments can be used to configure devc-ser8250, depending on the SBC and BIOS configuration. At the time of this writing the following initialization of the driver will work with the default configuration:

```
devc-ser8250 2f8,4 3f8,3 3e8,9 2e8,9
```

### Real Time Clock

QNX Momentics neutrino kernel supports the on-baord Real Time Clock. A battery is required (see hardware manual) to maintain the clock when the SBC is not powered.

## Console driver

No specialized driver is required for text consoles. For use with Photon, QNX Standard driver devg-geode.so provides full graphics support of the on-board video interface.

## USB

QNX Standard driver, devu-ohci, provides USB functionality on the PC-600 / PCS-620. See QNX provided documentation for more information.

## PC-104 and PC-104 Plus Bus and Adapters

Many drivers for possible PC-104 and PC-104 Plus expansion cards and adapters are available from QNX or from the expansion card vendor.

## Xpress AUDIO

The QNX Provided driver deva-ctrl-geode.so and audio manager io-audo provide support for the on-board audio subsystem. **However, the released version of the QNX pci server,** pci-bios**, is NOT compatible with the on-board audio subsystem.** This has been corrected by QNX, but was not available for release at the time of this writing. Contact QNX for support on this issue, and to obtain an updated pci-bios which supports audio on the Octagon Systems PC-600.

## $I^2C$ Bus Adapter

Octagon Systems provides the driver devc-i2c-gx1which provides access to the on-board $I^2C$ bus. This is required to use the devices connected to this bus as described below.

## Watchdog Timer

Support for the on-board watchdog timer is also provided with the devc-i2c-gx1 driver. (This is due to the hardware design of the $I^2C$ Bus and Watchdog interfaces, the Watchdog is not an $I^2C$ device.)

The watchdog is controlled through the device file /dev/watchdog.

The watchdog timer is initialized and enabled when the watchdog device file is opened. If the watchdog is enabled by the BIOS (configurable in the BIOS setup screen), the watchdog is running and timer starts at the point the BIOS hands off to the boot loader. In this configuration, the watchdog device file must be opened before this time out. Remember that the time spent starting the OS, and during the boot loader, must be considered.

## Serial EEPROM

Support for the serial EEPROM is provided with the devc-i2c-ee600 kernel module. This module requires the prior installation of devc-i2c-gx1.

The PC-600 and PCS-620 have 2048 bytes of EEPROM space available for user applications. This can be accessed through the file /dev/eeprom. The file behaves like an ordinary file, except that its length is fixed to 2048 bytes. NOTE: once opened, the EEPROM device contents are cached, and not written back to the device until closed or explicitly flushed.

## Temperature Sensor

The PC-600 and PCS-620 are equipped with a DS1775 temperature sensor. The kernel module devc-i2c-ds1775 provides access to the device. The devc-i2c-gx1 driver must be first be started in order to use this device.

There are two files, representing the two registers of the DS1775. For additional information on the temperature sensor, refer to the 5070 Single Board Computer User's Manual or the Dallas Semiconductor DS1775 SOT23 Digital Thermometer and Thermostat Product Review sheet

The file "DS1775_Config" can be read and written. Data transfer is a single byte, representing the config byte of the DS1775. For complete details on the configuration register, refer to the Dallas Semiconductor DS1775 SOT23 Digital Thermometer and Thermostat Product Review sheet.

The file "DS1775_Temp" is read only. Reading returns a single line, containing an ASCII decimal representation of the current temperature in degrees centigrade.

An example of temperature sensor code is provided in the appendix.

## Digital I/O

The PC-600 is equipped with 2 sets of 24 channels of digital I/O. The PCS-620 is equipped with one set of 24 channels of digital I/O. Access to these channels are provided by the driver devc-ezio.

The driver supports 24 channels of digital I/O. Since this modules will have to be loaded twice for the PC-600's two EZIO devices, the driver is executed twice, one instance for each EZIO device. Configuration options (IO base address, and ezio device number) are used to separate the two chips. Run "use devc-ezio" for more information.

The digital I/O is accessed through the device file /dev/ezio0 and /dev/ezio1. The file(s) may be accessed simultaneously up to 24 times from any number of processes.

Note: devc-ezio is not started automatically in the evaluation / demo compact flash shipped from Octagon Systems. You must start this driver before you can use

this interface. You may add this to the /etc/rc.init script for automatic startup if desired.

## Using the /dev/ezio*n* device file

The file must be opened with read/write access. Upon opening the file, the file descriptor is considered uninitialized. Any read or write operation to the file while uninitialized will result in an I/O ERROR (system call will return -EIO). The descriptor must be initialized with an ioctl() call which selects the desired I/O channel(s). Once initialized, reads / writes will return the current state or set the condition of the output driver respectively. The application may also use ioctl() calls to select the operating mode of the selected channels. Definitions required for using ioctl() calls are provided in ezio.h.

Valid devctl() calls for /dev/ezio:

| Request | Arguments |
|---|---|
| EZIOC_INIT | EZIOCINIT* |
| **Initialize file descriptor.** | |
| This call initializes a newly opened (uninitialized) file descriptor for /dev/ezio. The argument is a pointer to a structure of type EZIOCINIT. EZIOCINIT structure has two members (defined in ezio.h), the first is type int, named bit0, the second, also type int, named width.<br><br>The parameter "bit0" defines the I/O channel number for the lsb of all subsequent read / write operations. Valid range is 0 - 23.<br><br>The parameter "width" defines the number of consecutive I/O lines to be accessed by this file handle. Valid range is 1 to 8.<br><br>Return Values:<br>0 Success. I/O Channel is now initialized.<br>-EPERM Attempt to call EZIOC_INIT an a file descriptor already initialized. You may only initialize each descriptor once, successive attempts result in denial of permission.<br>-EACCESS Attempt to access I/O channels already in use by another file handle.<br>-EINVAL Invalid parameter. Either, both, or the combination of supplied parameters results in an attempt to access I/O channels that do not physically exist | |

| Request | Arguments |
|---|---|
| EZIOC_DDR | Unsigned char |
| **Set Data Direction Register** | |
| This call sets the direction of the bits assigned to the interface. A bit value of 1 set's the associated pin to an output, a value of 0 is input.<br><br>Return Values: | |

| | |
|---|---|
| 0 | Success. |
| -ENXIO | Attempt to call EZIOC_DDR on a file descriptor not yet initialized. You must initialize with EZIOC_INIT first. |

# *Appendix A - Programming examples*

The SBC card in the Development Kit comes with pre-installed example programs, located in the directory /example. The source code for these programs is provided on the CD under src/example. The source illustrates basics of accessing these features. Refer to the source code when running the examples to see how to use these features in your applications. An overview of these functions is provided in the previous sections.

## Digital I/O Examples

To assist the programmer in understanding the interface to the digital interface ports on the PC-600 and the one port on the PCS-620, examples are provided. There are many ways to use the provided interface, many more than can be completely covered in any simple example. These are intended as a basic functionality demonstration, and a starting point for developing your application.

Source for each is included on the OS Embedder CD for reference by the programmer.

The first example, "eziotest" uses the first I/O channels, transmits a simple walking 1 test pattern, and displays the current status of the interface pins on standard output. The output is formatted into 3 columns, of 8 bits. The first is bits 8 through 15, the second is bits 0 through 7, and finally the third is bits 16 through 23. In each bit, a "*" indicates a high level, and a - indicates a low (bit 1) level.

## Wdtest (watchdog timer)

The program "wdtest" demonstrates how to access the watchdog timer feature. Run the program /example/wdtest. The watchdog device file (/dev/watchdog) is opened, strobed periodically, then closed in a controlled (expected) manner which disables the watchdog. The program wdtest takes no parameters.

If the program is suspended, or killed, a reset will occur upon timeout. It is useful to note the different timeouts used if the function is stalled, without closing the watchdog device, or killed which results in an unexpected close of the file. (See watchdog chapter for timeout definitions.

## Testtemp (temperature sensor)

The program "testtemp" demonstrates access to the 5070's on-board temperature sensor. This program starts by reading the current system temperature, in ascii degrees C. Next, the configuration is changed to provide a more precise reading, and the temperature is re-read. Note that the temperature

conversion is slower in more precice modes. Finally the configuration register is set back to default settings.

The testtemp program does not take any parameters.